CHAPITRE 5

MÉTHODES DE RECHERCHE LOCALE

Sommaire

Introduction

Les méthodes de recherche locale sont des métaheuristiques à base de solution unique. La recherche se fait localement sur un ensemble de voisin de la solution actuelle.

Ce chapitre explique le principe de la recherche locale ainsi que la présentation de deux méthodes : La méthode de Recuit Simulé (Kirkpatrick *et al.*, 1983), et la méthode de recherche taboue (Glover, 1989).

1 Principe de la Recherche Locale

Vu l'explosion de l'espace de recherche d'un problème d'optimisation combinatoire, l'énumération peut être contournée par le fait de relier certaines solutions entre elles. Ainsi, à partir d'une solution, on peut en trouver une solution voisine.

Le principe des méthodes de recherche locale, appellées aussi méthodes de recherche de voisinage, se pose sur la construction d'un voisinage de solutions. Autrement dit, à partir d'une solution de départ S_0 on engendre une suite finie de solution S_n déterminées comme

plus proches. Généralement, la taille du voisinage est réduite. Ce principe est résumé dans l'algorithme 3.

Algorithme 3 : Algorithme général d'une méthode de recherche locale

Result : S : Solution
Données :
S_0 : Solution initiale;
S : Solution optimale;
S' : Solution voisine;
fit : réel;
début
Choisir S_0 une solution initiale;
$S \leftarrow S_0$;
$fit \leftarrow f(S)$;
tant que *($\exists S'$ dans le voisinage de S) et (acceptable(S'))* **faire**
$S \leftarrow S'$;
$fit \leftarrow f(S)$;
fin
fin

1.1 Transformation élementaire

La construction d'un voisinage est définie par des transformations élémentaires. Ces transformations sont des opérations qui permettent de changer une solution S en une autre solution voisine S'. Le voisinage $V(S)$ est l'ensemble de solutions *localement proche*, que l'on peut obtenir en appliquant à S une transformation locale élémentaire.

On cite quelques transformations élémentaires :

— **Complément :** Pour une solution codé en binaire, la transformation élémentaire est par exemple : remplacer un bit quelconque par son bit complémentaire :

$$000111 \longrightarrow 001111$$

— **Échange (Swap) :** Pour une solution codé en caractères (ou en binaires), permuter 2 caractères données :

$$ABCDE \longrightarrow AECDB$$

— **Insertion Décalage :** Pour une chaîne de caractères (ou binaires), choisir 2 positions i et j, mettre le caractère de la position j en position i et décaler les caractères à droite :

$$\mathbf{i=2}, \mathbf{j=5} : \quad ABCDEFG \longrightarrow AFBCDEG$$

2 Recuit Simulé (Simulated Annealing)

Le recuit simulé est une méthode de recherche locale qui simule le procédé de recuit en métallurgie. Ce processus alterne des cycles de refroidissement lents et de réchauffage (recuit) qui ont pour effet de minimiser l'énergie du matériau.

La méthode du recuit simulé, introduite par Kirkpatrick *et al.* (1983), exploite le voisinage et permet la redirection vers une solution voisine de moindre qualité avec une probabilité non nulle. Autrement dit, on n'améliore pas toujours la solution. Cette opération permet d'échapper les optima locaux.

L'algorithme 4 présente le fonctionnement de la méthode Récuit Simulé. Le paramètre T représente la température qui diminue tout en long de la résolution. Une probabilité est liée à la température et aux valeurs de la fonction de fitness de la solution actuelle et celle de la solution voisine. Cette probabilité définit l'acceptation de la nouvelle solution.

Algorithme 4 : Algorithme Recuit Simulé

Result : S : Solution
Données :
S_0, S et S' : Solution initiale, optimale et voisine;
$Voisin$: Liste de solutions;
T_0, T, fit : réels;
début
  Initialiser S_0 et T_0;
  $S \leftarrow S_0$;
  $T \leftarrow T_0$;
  $fit \leftarrow f(S)$;
  répéter
    Choisir aléatoirement $S' \in Voisin(S)$;
    $\Delta = f(S') - fit$;
    si $\Delta \leq 0$ **alors**
      $S \leftarrow S'$;
      $fit \leftarrow f(S)$;
    sinon
      Prendre p nombre aléatoire de $[0, 1]$;
      si $p < e^{-\Delta/T}$ **alors**
        $S \leftarrow S'$;
        $fit \leftarrow f(S)$;
      fin
    fin
    Mise à jour de la température T;
  jusqu'à *critère d'arrêt*;
fin

Discussion

- Si T est très élevée alors p est élevée ($-\Delta/T \curvearrowright 0$, $e^0 = 1$). On a de forte chance d'accepter la nouvelle solution (une dégradation de qualité).
- Si T est très petite alors p est très petite ($-\Delta/T \curvearrowright -\infty$, $e^{-\infty} = 0$). On a de forte chance de laisser la solution actuelle.
- Si Δ est un énorme alors la probabilité est petite.
- Plus la température T est petite plus la probabilité p est petite.

Dans la méthode du recuit simulé, les mécanismes d'intensification et de diversification sont contrôlés par la température. Une température élevée permet la diversification. Comme la température décroît, la recherche tend à s'intensifier vers la fin de l'algorithme.

3 Recherche Taboue

La recherche taboue a été introduite par Glover (1989). L'exploitation du voisinage permet de se déplacer de la solution courante vers son **meilleur** voisin. Ce dernier n'est pas forcément meilleur que la solution courante.

Cette méthode permet un déplacement d'une solution S vers la meilleure solution S' appartenant à son voisinage $V(S)$. L'algorithme 5 résume la procédure de la Recherche Taboue.

Le déplacement interdit, d'où vient le mot tabou, consiste au retour vers une solution récemment visitée. Pour cela, les solutions visitées sont temporairement interdites et stockées dans une liste taboue.

Une fois la liste taboue est remplie, la solution la plus ancienne est retirée (selon le principe d'une file d'attente). La taille de cette liste est un paramètre crucial qui affecte la résolution du problème. Elle peut être statique ou dynamique.

Discussion

- Si la liste taboue est courte :
  - Il y a moins d'interdictions.
  - On risque de n'exploiter que peu de distance (intensification).
  - Le risque de tourner en cycle, est plus grand.
- Si la liste taboue est longue :
  - Il y a d'avantage d'interdictions.
  - On parcourt de plus grandes distances (diversification).

Algorithme 5 : Algorithme Recherche taboue

Result : S : Solution

Données :

S_0, S et S' : Solution initiale, optimale et voisine;
$Voisin, ListeTaboue$: Liste de solutions;
T_0, T, fit : réels;

début

Initialiser $S_0, Voisin$ et $ListeTaboue$;
$S \leftarrow S_0$;
$fit \leftarrow f(S)$;
répéter
Choisir la meilleure $S' \in Voisin(S)$ tel que $S' \notin ListeTaboue$;
si S' *si elle est meilleure que* S **alors**
$S \leftarrow S'$;
fin
$Enfiler(ListeTaboue, S')$;
si $ListeTaboue$ *est pleine* **alors**
$Défiler(ListeTaboue)$;
fin
jusqu'à *critère d'arrêt*;

fin

- Le risque de cycles est réduit.

— Le comportement de l'algorithme dépend de :

- La longueur de la liste taboue.
- La taille du voisinage.

— Plus le voisinage est petit, moins la liste taboue a besoin d'être grande.

Dans la méthode de Recherche Taboue, les mécanismes d'intensification et de diversification sont contrôlés par la taille de la liste taboue : On fait de l'intensification, si on diminue la taille et on favorise la diversification dans le cas contraire.